Primjena strojnog učenja u predviđanju ponašanja potrošaća

Zekan, Marko

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Marko Zekan

PRIMJENA STROJNOG UČENJA U PREDVIĐANJU PONAŠANJA POTROŠAČA

ZAVRŠNI RAD

Varaždin, 2019.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Marko Zekan

Matični broj: 43411/14-R

Studij: Poslovni sustavi

PRIMJENA STROJNOG UČENJA U PREDVIĐANJU PONAŠANJA POTROŠAČA

ZAVRŠNI RAD

Mentor:

Izv. prof. dr. sc. Sandro Gerić

Varaždin, lipanj 2019.

Marko Zekan

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

_____

Sažetak

Rad je organiziran tako da je prvo objašnjena teorijska pozadina područja i uporaba metoda u stvarnom svijetu, a zatim je u praktičnom dijelu rada opisana implementacija obrađenih metoda. U teorijskom dijelu obrađena je tema uporabe modela strojnog učenja u poslovanju. Spomenute su uloge koje strojno učenje zauzima u raznim aspektima poslovanja kao što su: segmentacija klijenata, predviđanje poslovanja i sustavi preporuke. U prvom dijelu obrađena je i teorijska pozadina tri modela strojnog učenja korištena kasnije u radu, obrađeni modeli su: samo-organizirajuće (ESOM) mape, *k*-najbližih susjeda i *K*-sredina algoritam. U praktičnom dijelu rada prikazana je segmentacija kupaca i predviđanje ponašanja budućih i postojećih kupaca navedenim modelima strojnog učenja te analiza dobivenih rezultata.

Ključne riječi: strojno učenje, segmentacija klijenata, samo-organizirajuće mape, ESOM, predviđanje ponašanja potrošača, analiza podataka, K-najbližih susjeda, K-sredina algoritam, klasifikacija.

.

Sadržaj

1. Uvod

Ovaj završni rad bavit će se primjenom strojnog učenja u predviđanju ponašanja potrošača. U radu će biti prikazana teorijska pozadina teme, navedeni primjeri iz stvarnog svijeta, a zatim i praktična primjena nekolicine modela strojnog učenja.

U posljednjem desetljeću primjena umjetne inteligencije drastično je porasla u svim granama znanosti i industrije Novi načini obrade enormne količine podataka kroz modele koji mogu smisleno interpretirati te podatke nalaze svoje mjesto u industrijama kao što su medicina, avijacija i poslovanje. Jedan od pristupa koji je uvelike zaživio u posljednje vrijeme je strojno učenje. To je pristup analize podataka u kojem se kombinacijom statističkih i matematičkih modela te algoritama mogu pronaći obrasci u podacima. Temeljem pronađenih obrazaca u podacima se dalje mogu raditi predviđanja realizirana u obliku sustava preporuke, predviđanja potrošnje te općenito za marketinške svrhe.

Predviđanje ponašanja potrošača u ovom radu planira se ostvariti kroz par koraka. Prvi korak je segmentacija kupaca, ovdje će biti pokazana primjena specijalne vrste samo-organizirajuće mape kojom se planira postići odijeljivanje kupaca u nekolicinu zasebnih kategorija temeljem zajedničkih demografskih značajki. Nakon segmentacije kupaca biti će pokazani primjeri predviđanja ponašanja potrošača temeljem analize podataka i temeljem klasificiranja novog kupca. Praktični dio rada napravljen je kroz prizmu projekta iz znanosti o podacima (eng. *data science project*), što podrazumijeva korištenje najboljih praksi iz te industrije.

.

2. Strojno učenje u poslovanju

Ovo poglavlje rada analizirati će kakvu ulogu strojno učenje ima u poslovnom svijetu te kojim segmentima poslovanja se najviše koristi. Kasnije će biti opisano par područja gdje strojno učenje igra veliku ulogu.

Većina tvrtki danas barata s velikim količinama podataka. Podaci se dobivanju rudarenjem podataka iz mnogih izvora: rezultati poslovanja, marketinga, logistike, podaci o stanju tržišta. Količina i kompleksnost podataka raste kako i tvrtka raste, stoga je za izvlačenje korisnih informacije iz podataka potrebno koristit napredne tehnike (Yang, Deb, & Fong, 2011). Posljednjih godina te tehnike su većinom neka vrta algoritama strojnog učenja.

Želja da ti podaci budu maksimalno iskorišteni za dobivanje uvida u poslovanje, analize tržišta i pravljenja previđanja, tjera tvrtke da usvoje napredne tehnike strojnog učenja za analizu podataka. Forbes časopis 2018.godine navodi da se čak 51% ispitanih tvrtki izjasnilo kao rani usvojenici ili iskusni korisnici strojnog učenja u poslovanju (Bean, 2018). Ovakvo veliko usvajanje dovelo je do pronalaska mnogih primjena strojnog učenja na razne dijelove poslovanja kao i usavršavanje starih i razvoj novih metoda i modela.

Neka od područja poslovanja gdje se uvelike koriste tehnike strojnog učenja su („Top 8 Business Benefits of Machine Learning“, 2018):

- Segmentacija klijenata
- Preporuke proizvoda
- Financijska analiza
- Detektiranje slika
- Internet sigurnost
- Određivanje neželjene e-pošte
- Automatizacija unosa podataka
- Povećanje zadovoljstva klijenata

Sad će biti ukratko opisano korištenje tehnika strojnog učenja u područjima ponašanja klijenta i preporuke proizvoda.

2.1 Segmentacija klijenata

Jedan od načina predviđanja ponašanja potrošača je segmentiranjem tržišta. To je postupak odjeljivanje skupina klijenata u zasebne skupne prema nekim njihovim zajedničkim obilježjima. Segmentacija tržišta koristi se za alokaciju resursa i donošenje marketinških strategija. Pod segmentaciju klijenata ubraja se i svrstavanje budućih potencijalnih klijenata (Sari, Nugroho, Ferdiana, & Santosa, 2016), što je upravo jedan od ciljeva ovog rada.

Iako tradicionalne metode segmentacije tržišta daju dobre rezultate, povećanjem količine i kompleksnosti podataka postaju manjkave. Zato se danas za probleme klasifikacije i kategoriziranja uglavnom koriste umjetne neuronske mreže (Badea (Stroie), 2014). U ovom radu za segmentaciju klijenata biti će korištena umjetna neuronska mreža zvana - samo-organizirajuća mapa.

2.2 Sustavi preporuke proizvoda

Sustavi preporuke (eng. *recommendation systems/recommendation engines*) su programi koji pokušavaju predvidjeti korisnikove interese (Sublett, 2018). Sustavi se dijele na dvije podskupine, prva skupina preporučuje proizvode temeljem sličnosti proizvoda s već kupljenim proizvodima, a druga temeljem korisnikovih preferencija i interakcija. Također, postoje i sustavi preporuke koji kombiniraju obje metode.

Sustavi preporuke uglavnom rade na način da iz baze podataka prema nekom kriteriju biraju proizvode koje će ponuditi korisniku. Kako na ekran ne mogu stati svi proizvodi, bitno je imati sustav preporuke koji može što bolje odabrati koje će proizvode koje će prikazati korisniku (Pazzani & Billsus, 2007). Kriterij odluke temelji se na jedan od dva načina spomenuta u prijašnjem paragrafu ili kombinacijom oba.

U ovom radu biti će pokazana preporuka proizvoda temeljem podataka o prijašnjoj kupnji. Zamisao je da se nakon segmentacije, kupcima preporuče proizvodi koje su kupovali samo članovi njihovog segmenta, odnosno kategorije. Na ovaj način moguće je napraviti preporuku za novog kupca koji još nije obavio prvu kupnju.

3. Modeli strojnog učenja

Kao što je već spomenuto, za postizanje zadanog cilja, u ovom radu korišteno je više modela strojnog učenja. U ovom poglavlju biti će opisana teorijska pozadina modela samo-organizirajuće mape, k-sredina i k-najbližih susjeda.

3.1 Samo-organizirajuće mape

Samo-organizirajuće mape (eng. *Self-Organizing Maps, SOM*) su nenadziranih modeli strojnog učenja, spadaju pod tip umjetnih neuronskih mreža, te se često koriste za vizualizaciju i kategoriziranje (eng. *clustering*) visoko-dimenzionalnih skupova podataka. SOM, odnosno Kohonenove mape, predložio je finski profesor Teuvo Kohonen 1982. godine u radu „*Self-Organized Formation of Topologically Correct Feature Maps*“ koji ga je uvelike proslavio u području umjetne inteligencije i biološki-inspiriranih sustava (Asan & Ercan, 2012). Samo-organizirajuće mape koriste se u pronalaženju obrazaca među podacima, robotici, telekomunikacijama, optimizaciji i poslovanju.

SOM mreže sastoje se od dva sloja neurona gdje je svaki neuron prvog sloja povezan sa svim neuronima drugog sloja. Prvi, odnosno ulazni sloj, sadrži broj neurona jednak broju dimenzije skupa. Na primjeru ovog rada, dimenzija slupa je šest zato što će se predviđanje radit temeljem šest varijabli o svakom kupcu, stoga i broj ulaznih neurona iznosi šest. Izlazni sloj, Kohonenov sloj ili SOM sloj mreže sastoji se od većeg unaprijed određenog broja neurona, koji su organizirani u mapu tako da predstavljaju dvodimenzionalni prikaz više-dimenzionalnog skupa podataka (Asan & Ercan, 2012). Neuroni na izlaznom sloju napravljeni su u obliku vektora iste dimenzije kao i ulazni sloj. Dakle, u ovom primjeru, svaki neuron izlaznog sloja sadržavao bi šest vrijednosti.

Cilj SOM mreže je grupirati sve instance skupa podataka prema njima najsličnijem neuronu na izlaznom sloju mreže. Na taj način se svaka instanca podataka može smjestiti u jednu zasebnu grupu u kojoj se nalaze njoj slične instance. U tradicionalnim jednostavnim SOM mrežama broj neurona na izlaznom sloju otprilike je jednak broju grupa u koje želimo smjestit podatke (Ultsch & Morchen, 2005).

Algoritam SOM mreže odvija se kroz šest koraka (Ghany & Solano, 2012):

1. **Inicijalizacija utega** (eng. *weight initialization*) – svaka vrijednost vektora u izlaznom sloju mreže inicijalizira se nasumičnom vrijednošću.

2. **Odabir nasumičnog retka iz skupa** – odabire se nasumičan redak iz skupa podataka te se daje mreži na obradu.
3. **Pronalazak pobjedničkog neurona** (eng. *Best Matching Unit, BMU*) – procjenjuje se vrijednost vektora svakog neurona kako bi se pronašao vektor čija vrijednost je najsličnija odabranom retku iz 2. koraka. Sličnost vektora računa se prema euklidskoj udaljenosti od ulaznog retka. Neuron čija vrijednost je najbliža ulaznom retku je pobjednički neuron, odnosno BMU.
4. **Određivanje susjedstva pobjedničkog neurona** – pronalazi se svaki susjedni neuron koji se nalazi unutar zadanog radijusa oko BMU-a. Radijus susjedstva računa se prema vrijednosti zadanoj pri inicijalizaciji modela. Običaj je da se radijus susjedstva smanjuje svakom iteracijom. Time se postiže kreiranje bolje odijeljenih grupa na mapi.
5. **Ažuriranje susjednih neurona** – sve vrijednosti u vektorima susjednih neurona ažuriraju se prema unaprijed određenoj funkciji. Što se neuron nalazi bliže BMU to će njegov vektor biti više promijenjen. Ovim korakom se također postiže koherentnije grupiranje podataka.
6. **Koraci 2**.- **5. ponavljaju se zadan broj iteracija.**

Implementacija jedne vrste SOM mreže nalazi se u praktičnom dijelu rada. SOM mreža koristit će se za interpretiranje podataka o kupcima na način da se podaci predstave neuronima mreže. Ovime se postiže grupiranje neurona temeljeno na razlikama među individualnim instancama skupa podataka, koje će na kraju činit tražene kategorije kupaca. Neuroni će se zatim mogu odijeliti u kategorije vizualno ili korištenjem nekog algoritma.

3.2 Algoritam K-sredina

Algoritam grupiranja K-sredina (eng. *K-Means Clustering algorithm*) je nenadzirani algoritam strojnog učenja kojeg je predložio J. MacQueen 1967. godine u svom radu „*Some Methods for classification and Analysis of Multivariate Observations*“. Algoritam se koristi za automatsko odijeljivanje skupa podataka u predodređen broj grupa *k* (Wagstaff, Cardie, Rogers, & Schroedl, 2001).

Algoritam k-sredina je prilično jednostavan, odvija se u samo dva koraka sve dok se ne zadovolji zadani kriterij terminiranja (Mirošević, 2016). Pri inicijalizaciji algoritma, na nasumičnim pozicijama u prostoru kreira se *k* broj predstavničkih točaka. Te točke služit će kao mjesta prema kojima će ostale točke konvergirati, čime će biti dobivene odijeljene grupe. Cilj algoritma je smjestit sve instance skupa podataka u grupe s njima sličnim drugim

instancama. Algoritam k-sredina se zbog svoje jednostavnosti upotrebljava u mnogim disciplinama i u mnoge svrhe (Raghupathi, 2018). Neke od primjena modela su: klasificiranje dokumenata, segmentiranje klijenata, otkrivanje prevara u osiguranju te profiliranje cyber-kriminalaca.

Neke od prednosti k-sredina algoritma su njegova jednostavnost, u smislu da je lako interpretirati dobivene rezultate za razliku od nekih drugih algoritama strojnog učenja. Također algoritam k-sredina nije jako računalno zahtjevan, što ga čini pogodnim i za obradu velikih skupova podataka. Najveća mana modela k-sredina je što se broj kategorija *k* mora odrediti prije pokretanja algoritma. Još jedna mana je što kvaliteta odijeljivanja podataka ovelike ovisi o nasumično zadanim predstavničkim točkama pri inicijalizaciji algoritma. Također, nasumično biranje tako važnog dijela algoritma ima za posljedicu teško repliciranje dobivanih rezultata (Darrin, 2016).

Postupak odvijanja algoritma u dva koraka (Mirošević, 2016):

1. **Korak pridruživanja** – nakon nasumične inicijalizacije predstavničkih točaka (eng. *centroids*) svaka točka iz skupa podataka pridružuje se njoj najbližoj predstavničkoj točki. Udaljenost između instanci podataka iz skupa i predstavničkih točaka računa se kao euklidska udaljenost u prostoru.
2. **Korak prepravljanja** – jednom kada su udaljenosti izračunate, predstavničke točke se prepravljaju tako da opet budu pozicionirane u sredini između svih točaka te grupe. Na ovaj način se svakom iteracijom kreiraju grupe kojima se u sredini nalazi predstavnička točka. Nove koordinate predstavničke točke određuju se tako, da se izračuna medijan (srednja vrijednost) svih udaljenosti točaka unutar grupe sa njenom predstavničkom točkom. Iz toga proizlazi ime algoritma - k-*sredina*.

Koraci algoritma ponavljaju se sve dok se vrijednost ciljne funkcije ne prestane smanjivati. Drugim riječima, koraci se ponavljaju sve dok koordinate predstavničkih točaka ne budu iste više iteracija za redom, što znači da se grupe ne mogu bolje odijeliti. Time je završen proces kategorizacije.

U praktičnom dijelu rada, algoritam k-sredina bit će korišten za određivanje optimalnog broja grupa u koje svrstati kupce. Također, bit će korišten za odijeljivanje neurona SOM mreže u kategorije.

3.3 Algoritam k-najbližih susjeda

Algoritam k-najbližih susjeda (eng. *k-nearest neighbor, k-NN*) jedan je od jednostavnijih algoritama u strojnom učenju. K-NN spada pod kategoriju nadziranih algoritama te se koristi uglavnom za problem klasifikacije. Algoritam radi pod pretpostavkom da se slični podaci nalaze vrlo blizu jedni drugima u prostoru (Harrison, 2018). Takva pretpostavka omogućava da se još neklasificirani podaci klasificiraju temeljem sličnosti s neposrednim „susjedima“ u prostoru. Odnosno, ako je neklasificirani podatak u prostoru okružen podacima označenim, na primjer, klasom „A“, tada će i novi podatak biti klasificiran u klasu „A“.

Računanje sličnosti jedne instance skupa podataka s nekom drugom instancom podatka radi se računanjem njihove udaljenosti u prostoru. Ovakvim pristupom svaka instanca podatka smatra se točkom u prostoru između kojih se mjeri udaljenost, što su točke bliže jedna drugoj to su sličnije Za računanje udaljenosti najčešće se koristi euklidska udaljenost u prostoru prema općenitoj formuli („Euclidean Distance“, 2005):

$$d(p_1, p_2) = \sqrt{\sum_{i=1}^{v} (p_{1i} - p_{2i})^2}$$

Pri čemu je $d(p_1, p_2)$ udaljenost između dvije točke u prostoru p_1 i p_2, v je broj instanci skupa podataka.

Algoritam k-najbližih susjeda spada pod algoritme lijenog učenja (eng. *lazy learning*). Većina modela strojnog učenja oslanja se na treniranje modela prije donošenja odluke za nove podatke, pri čemu se za treniranje modela koristi cijeli skup podataka. Ovakvim pristupom pokušava se izgraditi sposobnost generalizacije modela temeljem čega će donositi odluke za nove podatke. Alternativa tome je korištenje pristupa lijenog učenja, to jest pristup u kojem se umjesto temeljem cijelog skupa podataka odluka donosi temeljem manjeg podskupa podatka (Galván, Valls, García, & Isasi, 2011). K-NN algoritam spada pod algoritme lijenog učenja zato što za donošenje odluke (klasificiranja nove instance podatka) koristi samo manji broj (*k*) neposrednih susjeda u prostoru.

K-NN model se zbog svoje jednostavnosti može primijeniti na širok spektar problema. Neka od grana gdje se K-NN algoritam koristi su: bankarstvo – za odluku o izdavanju kredita i pozajmica klijetima, politika – predviđanje kome će glasač dati glas ili hoće li uopće glasati. Još neke primjene uključuju prepoznavanje rukopisa te klasifikacija slika i videa (Bronsthein, 2017).

Algoritam k-NN klasifikacije odvija se na sljedeći način (Harrison, 2018):

1. **Određivanje K broja susjeda** – potrebno je odrediti koliko susjeda nekoj točci mora biti slično da bi se točka klasificirala kao dio tog pod-skupa. Ovaj postupak najčešće se radi tako da se klasifikacija vrši na istom skupu podataka više puta, ali svaki put s različitim brojem *k* (recimo od *k*=1 do *k*=30). Pri svakoj iteraciji mjeri se točnost algoritma te se na kraju uzima onaj *k* za koji je točnost bila najveća.
2. **Računanje udaljenosti za svaku točku** – računa se udaljenost između točke koja se želi klasificirat i svih ostalih točaka u ravnini. Zatim se uzima *k* najmanjih udaljenosti (odnosno *k* najbližih točaka) i provjerava koja je njihova oznaka (eng. *label*).

Novoj instanci podatka biti će dodijeljena ista klasa kojoj pripada najviše njenih susjeda. K-NN algoritam koristiti će se u praktičnom dijelu rada, za klasificiranje novih kupaca. Klase prema kojima će algoritam raditi biti će dobivene kategorizacijom skupa podataka temeljem SOM modela i K-sredina algoritma.

4. Predviđanje ponašanja potrošača putem strojnog učenja – praktični dio

Ovaj dio završnog rada pratiti će implementaciju modela strojnog učenja, analizu dobivenih rezultata. Programski dio izrađen je u obliku IPython bilježnice te se može pogledati ili preuzeti sa sljedeće Github poveznice - https://github.com/marzekan/Zavrsni-rad. Osim izvornog programskog koda, na poveznici se mogu pronaći i spremljeni modeli trenirani za potrebe ovog rada.

Za izradu praktičnog dijela rada korišten je programski jezik Python 3.6 (https://www.python.org) unutar *Jupyter Notebook* (https://jupyter.org/) okružja te nekolicina Python modula koji uvelike olakšavaju posao analize podataka i izrade inače kompleksnih algoritama. Najvažniji od korištenih modula su:

- ***Pandas*** - korišten za lakše formatiranje i analizu podataka (https://pandas.pydata.org/).
- ***Somoclu*** - sadrži metode i klase za izgradnju samo-organizirajućih mapa (https://somoclu.readthedocs.io/en/stable/).
- ***Scikit-learn*** - poznati modul koji sadrži mnogo modela strojnog učenja i metoda za rad s njima (https://scikit-learn.org/stable/).
- ***Matplotlib i Seaborn*** - moduli za izradu dijagrama (https://matplotlib.org, https://seaborn.pydata.org)

Glavna zamisao ovo dijela rada je primijeniti tehnike strojnog učenja kroz prizmu *data science* projekta na segmentaciju kupaca, analizu segmenata te predviđanje budućeg ponašanja kupaca. Za segmentaciju kupaca koristiti će se specijalna vrsta samo-organizirajuće mape – ESOM (eng. *Emerging Self-Organizing Map*) i algoritam k-sredina. U svrhu pravljenja predviđanja za nove kupce biti će korišten *K*-najbližih susjeda (eng. *K-Nearest Neighbors,* K-NN) klasificirajući algoritam. Ideja je nenadziranim (eng. *unsupervised*) algoritmom podijeliti postojeće kupce u određen broj segmenata, odnosno kategorija (eng. *clusters*), tako da se kasnije može koristiti nadzirani (eng. *supervised*) algoritam za predviđanje ponašanja novih kupaca.

Temeljan dio svakog dobrog *data science* projekta je dobar skup podataka. U slučaju ovog rada, skup podataka se zove *Black Friday* te je preuzet besplatno sa *Kaggle* web stranice (https://www.kaggle.com/mehdidag/black-Friday) .

U području rada s podacima postoje već definirani koraci za provođenje projekata, nastali praćenjem najboljih praksi iz znanosti i industrije (Géron, 2017, str. 56). Preporučeni koraci prilagođeni su za potrebe ovoga rada te će naredni praktični dio rada biti proveden na sljedeći način:

1. **Početna analiza podataka (eng. *data exploration*)**

   U prvom koraku biti će analizirana i objašnjena struktura skupa podataka te grafički prikazane učestalosti pojedinih podataka, trendova i ostalih odnosa među podacima.

2. **Odabir modela**

   Temeljem informacija dobivenih analizom u prvom koraku postavljamo problem, zatim se bira jedan ili više modela strojnog učenja kojim će se riješili taj problem.

3. **Čišćenje podataka i priprema podataka za samo-organizirajuću mapu (eng. *data cleaning*)**

   Priprema podataka za strojno učenje, odnosno popunjavanje podataka koji fale, skaliranje, normalizacija i općenita standardizacija skupa podataka.

4. **Primjena samo-organizirajuće mape na segmentaciju kupaca**

   U ovom koraku obradit će se odabir hiper-parametara te modeliranje i treniranje mape na skupu podataka. Zatim, analiza topologije mape te vraćanje dobivenih rezultata u početni skup podataka.

5. **Analiza dobivenih rezultata**

   Analiza odnosa među podacima unutar kategorija dobivenih samo-organizirajućom mapom.

6. **Predviđanje za nove i postojeće kupce**

   Korištenje dobivenih rezultata (kategorija) za klasifikaciju novog kupca putem KNN algoritma te odabir proizvoda za preporuku postojećem kupcu.

4.1 Početna analiza podataka

Black Friday skup podataka korišten u ovom radu sadrži podatke o 5.891 kupcu i njihovoj kupovini na dan Crnog petka neke trgovačke kuće. Skup sadrži podatke više kupovina od svakog kupca stoga broj redaka iznosi 537.577, opisanih kroz dvanaest stupaca. U stupcima se redom nalaze sljedeći podaci:

- *User_ID* - jedinstveni identifikator kupca, cjelobrojnog tipa podatka.
- *Product_ID* - jedinstveni identifikator proizvoda, *string* tipa podatka.
- *Gender* - spol kupca, poprima jednu od dvije *string* vrijednosti „M“ za muškarce, „F“ za žene.
- *Age* - dob kupca, određena jednom od sedam kategorija. Kategorije su: '0-17', '18-25', '26-35', '36-45', '46-50', '51-55', '55+'.
- *Occupation* - kategorija zanimanja kojom se kupac bavi. Postoji dvadeset jedno zanimanje, označeno brojevima: 0 – 20.
- *City_Category* - kategorija grada u kojem kupac živi. Postoje tri kategorije grada označene slovima. A, B, C.
- *Stay_In_Current_City_Years* - koliko godina kupac živi na trenutnoj lokaciji. Razlikuje se pet kategorija godina: '0', '1', '2', '3', '4+'.
- *Marital_Status* - bračno stanje kupca. Cjelobrojna vrijednost, iznosi 0 za kupce koji nisu u braku te 1 za one koji jesu u braku.
- *Product_Category_1* - kategorija kojoj kupljeni proizvod pripada. Kategorije su označene cijelim brojem od 1 do 18.
- *Product_Category_2* - kategorija kojoj proizvod pripada. Proizvod može pripadati u više kategorija, također određenih cijelim brojevima od 1 do 18.
- *Product_Category_3* - kategorija kojoj proizvod pripada. Proizvod može pripadati u više kategorija, također određenih cijelim brojevima od 1 do 18.
- *Purchase* - iznos novca potrošen u jednoj kupovini (u američkim dolarima).

Neka dodatna opažanja mogu biti dobivena pozivom `info()` metode nad skupom podataka na sljedeći način:

```
skup_podataka.info()
```

Rezultat poziva metode prikazan je na slici 1.

```
RangeIndex: 537577 entries, 0 to 537576
Data columns (total 12 columns):
User_ID                     537577 non-null int64
Product_ID                  537577 non-null object
Gender                      537577 non-null object
Age                         537577 non-null object
Occupation                  537577 non-null int64
City_Category               537577 non-null object
Stay_In_Current_City_Years  537577 non-null object
Marital_Status              537577 non-null int64
Product_Category_1          537577 non-null int64
Product_Category_2          370591 non-null float64
Product_Category_3          164278 non-null float64
Purchase                    537577 non-null int64
dtypes: float64(2), int64(5), object(5)
memory usage: 49.2+ MB
```

Slika 1. Informacije o skupu podataka (autorski rad)

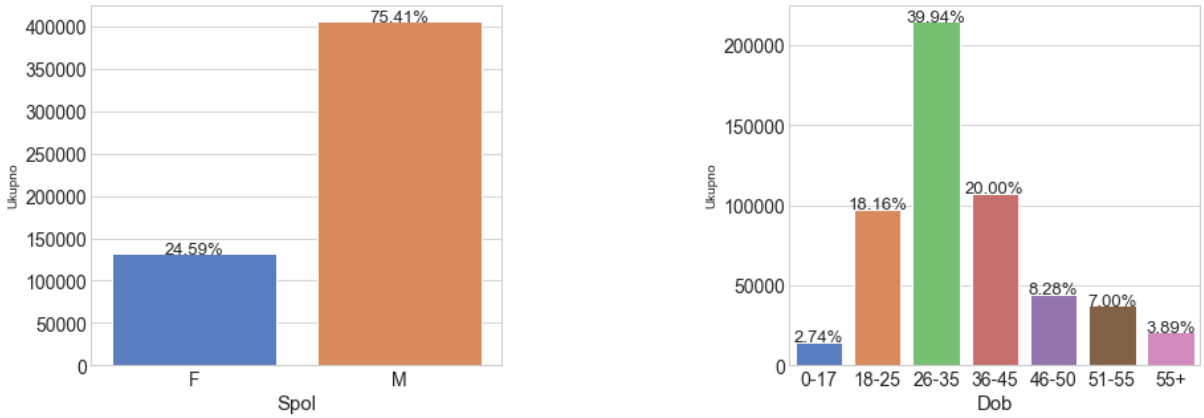
Iz slike X možemo primijetiti da su podaci različitih tipova podataka, da se ne sastoje od isključivo numeričkih vrijednosti te da dva stupca sadrže *null* vrijednosti (stupci *Product_Category_1* i *Product_Category_2*). Sva nabrojana opažanja predstavljaju problem za analizu algoritmima strojnog učenja, stoga je potrebno detaljno očistit podatke prije korištenja modela. Na slici 2. nalazi se uzorak prvih pet redaka skupa podataka.

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Produ
0	1000001	P00069042	F	0-17	10	A	2	0	3	NaN	
1	1000001	P00248942	F	0-17	10	A	2	0	1	6.0	
2	1000001	P00087842	F	0-17	10	A	2	0	12	NaN	
3	1000001	P00085442	F	0-17	10	A	2	0	12	14.0	
4	1000002	P00285442	M	55+	16	C	4+	0	8	NaN	

Slika 2. Uzorak skupa podataka (autorski rad)

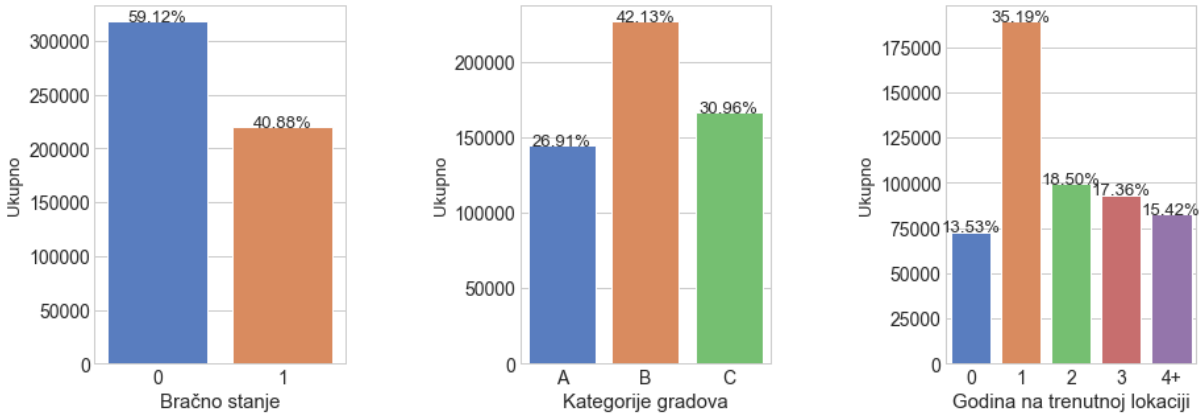
Sljedećim dijagramima prikazana je zastupljenost spolova i dobnih kategorija u skupu podataka. Iz grafikona 1. možemo vidjeti da su većina kupaca muškarci (75.41%) te

da većina kupaca pripada dobnim kategorijama: '18-25' (18.16%), '26-35' (39.94%) i '36-45' (20%).



Graf 1. Udio kupaca prema dobi i spolu u skupu podataka (autorski rad)

Grafikon 2. prikazuje redom, zastupljenost bračnog stanja među kupcima, kategorije gradova u kojima žive te koliko godina žive na sadašnjoj lokaciji. Vidljivo je da većina kupaca nije u braku (59.12%), većina živi u "B" kategoriji grada (42.13%) te na sadašnjoj lokaciji uglavnom žive godinu dana (35,19%).

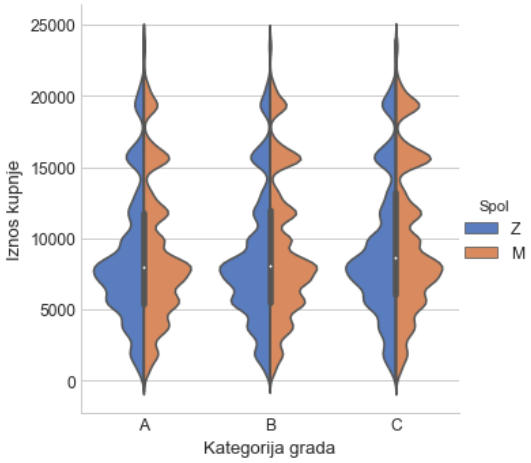


Graf 2. Udio kupaca prema bračnom stanju, kategoriji grada te godini stanovanja na sadašnjoj lokaciji (autorski rad)

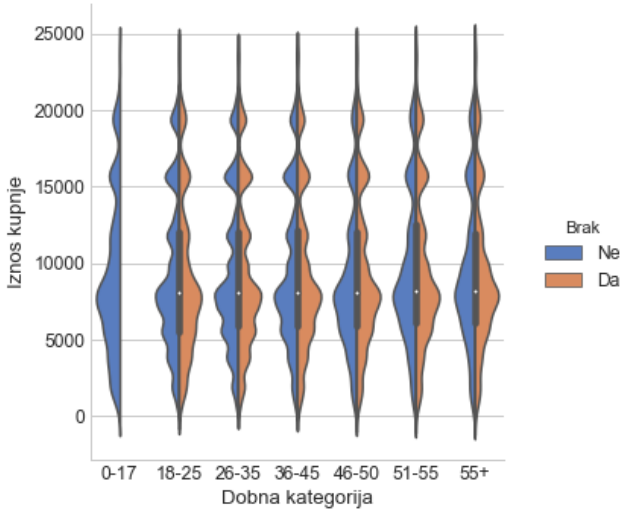
Iz grafikona 3. vidljivo je da je prosječna potrošnja u kategoriji grada "C" nešto veća u odnosu na ostale kategorije, gdje je prosječna potrošnja skoro identična po svim kriterijima. Najviše iznosa kupovine nalazi se između pet i deset tisuća dolara, skoro ravnomjerno

raspodijeljeni među muškarcima i ženama, dok muškarci više troše u višim cjenovnim razredima (petnaest do dvadeset tisuća dolara), specifično u razredu od petnaest do dvadeset tisuća dolara.

Grafikon 4. prikazuje potrošnju prema dobnim kategorijama i bračnom stanju. Uočljivo je da se medijan (bijela točka u unutrašnjosti grafikona) potrošnje ne mijenja prema dobi niti prema bračnom stanju. Struktura potrošnje skoro jednaka prijašnjem grafikonu, ipak, prva dobna kategorija (0-17) pokazuje ravnomjerniju potrošnju u usporedbi s ostalim kategorijama.

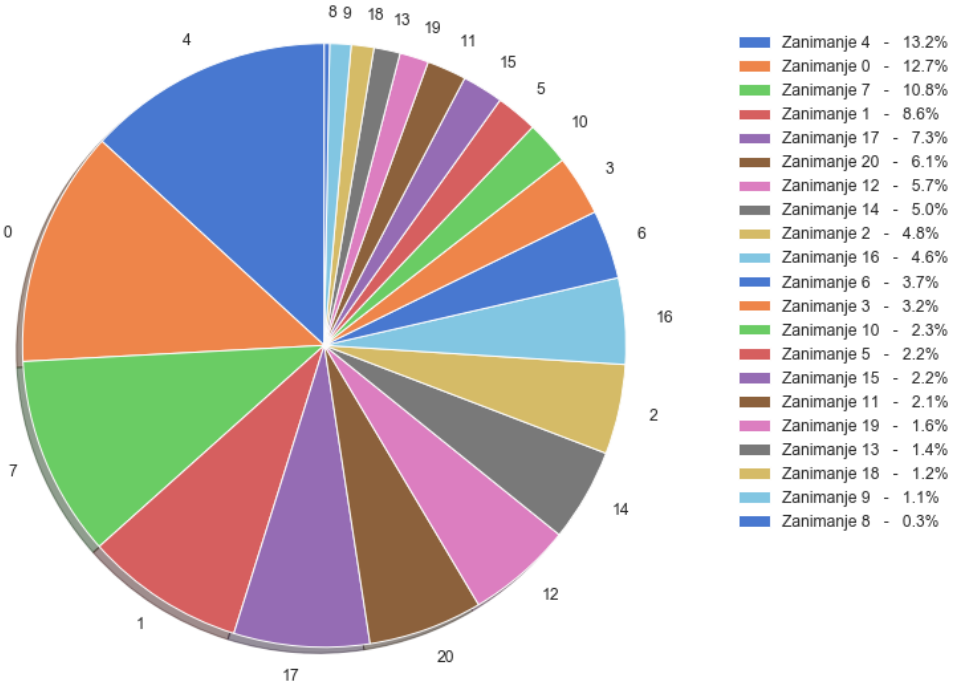


Graf 3. Distribucija kupaca prema potrošnju, kategoriji grada i spolu (autorski rad)



Graf 4. Distribucija kupaca prema dobnoj kategoriji, bračnom stanju te iznosu kupnje (autorski rad)

Sljedeći grafikon prikazuje udio zanimanja sortiran od najčešćeg. Iz grafikona je vidljivo da je zanimanje broj '4' sa udjelom od 13.2% najzastupljenije, dok su zanimanja '0' i '7' na drugom i trećem mjestu sa 12.7% i 10.8% respektabilno.



Graf 5. Udio zanimanja kupaca (autorski rad)

4.2 Odabir modela

Sljedeći korak je odabir jednog ili više modela strojnog učenja. Modeli se biraju prema tipu problema koji je potrebno riješiti, stoga je prvo potrebno postaviti problem. Neki do najčešćih tipova problema u strojnom učenju su (Brownlee, 2013):

- **Klasifikacija** (eng. *classification*).
- **Regresija** (eng. *regression*).
- **Grupiranje** (eng. *clustering*).
- **Izvlačenje asocijativnih pravila** (eng. *rule extraction*).

Obzirom da se u ovom radu nastoji postići predviđanje za nove i za postojeće kupce, iz skupa podataka biti će korišteni samo podaci vezani uz demografiju kupaca (dob, spol itd.). Ovime se želi postići mogućnost predviđanja čak i prije nego kupac napravi prvu kupovinu kao i mogućnost prepoznavanja specifičnih podskupina kupaca za marketinške svrhe. Pretpostavka je da su demografski podaci spremljeni prilikom kreiranja računa za Internet trgovinu ili anketom.

Klasificiranje novog kupca bilo bi lako kada bi svi postojeći kupci u skupu već pripadali nekoj kategoriji i kad bi bili označeni prema njoj (eng. *labeled*). No kako tome nije tako, potrebno je prvo podijeliti postojeće kupce u kategorije, stoga je prvi problem, problem kategoriziranja/grupiranja. Jedan način za kategorizirati podatke je upotrebom algoritma kategoriziranja (eng. *clustering algorithm*) koji sami pronalaze sličnosti i obrasce među podacima i svrstavaju ih i u određen broj kategorija odnosno grupa. Na taj način je moguće podijeliti sve postojeće kupce u određeni broj kategorija tako da se u svakoj kategoriji nalaze samo kupci koji imaju neke zajedničke osobine. Nakon što algoritam podijeli postojeće kupce u grupe, lako je primijeniti algoritam klasificiranja da klasificira nove kupce temeljem ranije dobivenih oznaka. Jednom kada je novi kupac označen kao član neke kategorije, moguće je napraviti razne analize temeljem postojećih podataka kako bi odredili što će novi kupac kupiti ili koliko će potrošiti. Drugi problem je stoga, problem klasifikacije.

Postoji mnogo vrsta algoritama kategoriziranja, no za većinu njih treba unaprijed odrediti broj kategorija (K-sredina algoritam na primjer). Sukladno cilju ovoga rada broj kategorija neće biti unaprijed određen već će biti ostavljeno odabranom algoritmu da pronađe koliko kategorija ima smisla za ovaj specifičan skup. U tom slučaju, izbor algoritama je manji.

Neki od algoritama koji mogu kategorizirat podatke bez zadanog broja kategorija su:

- Hijerarhijsko kategoriziranje (eng. *hierarchical clustering*)
- DBSCAN
- Samo-organizirajuće mape

Svaki od navedenih algoritama ima svoje prednosti i mane, ne postoji niti jedan algoritam koji može riješiti bilo koji problem. Stoga je vrlo bitno odabrati najbolji mogući algoritam za specifičan problem koji je potrebno riješiti.

Algoritam hijerarhijskog kategoriziranja za cilj ima napraviti hijerarhijsku strukturu od danih podataka u obliku stabla. Postoji više implementacija ovog algoritma, no svi pristupi za donošenje odluka koriste matricu sličnosti građenu prema udaljenostima među točkama u prostoru (Kilitcioglu, 2018). Prema (Bhagat, Kshirsagar, Khodke, Dongre, & Ali, 2016) neke od mana hijerarhijskog kategoriziranja su: loš rad s visoko-dimenzionalnim podacima, troše

mnogo računalnih resursa pri obradi velikih količina podataka te su općenito prilično statične prirode, bez mogućnosti prilagodbe promjenjivim podacima. Iako skup podataka korišten u ovom radu nije jako velik, inače skupovi podataka o klijentima znaju biti ekstremno veliki, milioni redaka te tisuće stupaca, stoga ovaj model nije dobar za prikaz primjera segmentacije klijenata.

DBSCAN (eng. *Density-Based Spatial Clustering of Applications with Noise*) algoritam radi vrlo dobro za skupove podataka koji su sadržani od dijelova velike gustoće razdvojenih mjestima male gustoće, tako da ovaj algoritam ne zahtjeva unaprijed određen broj kategorija jer ih može sam prepoznati prema broju gustih područja u ravnini. Velika mana DBSCAN algoritma je da ne radi dobro nad podacima ujednačene gustoće (Dang, 2015). Kasnije će biti pokazano da je skup podataka korišten u ovom radu prilično je ujednačene gustoće, što čini DBSCAN algoritam nekompetentnim za rješenje problema kategorizacije.

Od navedenih modela, samo-organizirajuće mape jedine spadaju u modele umjetnih neuronskih mreža. Način na koji SOM procesira podatke (str. 4), čini ju pogodnom za rješenje problema kategorizacije u ovom slučaju. Također, primjena specijalne vrste samo-organizirajuće mape zvane ESOM na problem kategoriziranja pokazuje bolje rezultate u usporedbi s drugim modelima (Ultsch, 2005).

4.3 Čišćenje i priprema podataka za samo-organizirajuću mapu

Čišćenje podataka izrazito je bitan korak kod projekata koji uključuju strojno učenje, a odnosi se na postupke nadopunjavanja i odstranjivanja podataka koji nedostaju u skupu, kao i općenita standardizacija vrijednosti u skupu podataka (Le, 2019). U ovom poglavlju biti će odrađena selekcija značajki, standardizacija podataka i skaliranje podataka kako bi na kraju podaci bili spremni za SOM model.

4.3.1 Odabir značajki

Odabir značajki (eng. *feature selection*) je postupak u kojem se ručno ili automatski biraju varijable koje će se koristit za daljnja predviđanja. Kod velikih skupova podataka s mnogo dimenzija potrebno je odabrati samo bitne značajke za postizanje zadanog cilja. Odabir bitnih značajki može pomoći pri dostizanju željenog rezultata vremenski i točnošću, dok ne činjenje toga može negativno utjecati na ishod učenja (Shaikh, 2018).

Ranije je rečeno da je cilj ovog rada predviđanje ponašanja kupaca temeljem demografskih značajki. Obzirom da *Black Friday* skup podataka sadrži po nekoliko desetaka instanci kupnje za svakoga kupca, potrebno je izdvojiti samo podatke o kupcima bez podataka o kupnji (identifikator kupljenog proizvoda, iznos potrošnje, itd.).

Iz originalnog skupa podataka uzimaju se stupci: identifikator kupca, dob, spol, zanimanje, godina na trenutnoj lokaciji, bračno stanje i kategorija grada. Identifikacijski kod kupca nije demografska vrijednost te će biti kasnije uklonjen, ali trenutno služi kao provjera valjanosti postupaka čišćenja. Na početku poglavlja pokazano je da dva stupca, *Product_Category_2* i *Product_Category_3* sadrže prazne vrijednosti redaka. Kako ti stupci neće biti korišteni za treniranje modela, nema potrebe popunjavati njihove prazne retke. Uzorak modificiranog skupa podataka nalazi se na slici 3.

	User_ID	Age	Gender	Occupation	Stay_In_Current_City_Years	Marital_Status	City_Category
0	1000001	0-17	F	10	2	0	A
1	1000001	0-17	F	10	2	0	A
2	1000001	0-17	F	10	2	0	A
3	1000001	0-17	F	10	2	0	A
4	1000002	55+	M	16	4+	0	C

Slika 3. Uzorak skupa sa probranim značajkama (autorski rad)

4.3.2 Standardizacija podataka

Sljedeći korak pripreme podataka je pretvaranje ne-numeričkih tipova podataka u numeričke tipove podataka. No prije toga treba učinit još neke preinake na skupu. Odstranjivanjem stupaca vezanih uz kupovinu u skupu su ostali samo demografski podaci o kupcima koji su isti u svakoj instanci kupnje, stoga je potrebno maknuti duplikate redaka. Na slici 4., prema identifikatoru kupca može se vidjeti kako su duplikati redaka odstranjeni. Na kraju se još odstranjuje i stupac sa identifikacijskim kodom kupca te se dobiva skup podataka sa 5.891 retkom i 6 stupaca.

	User_ID	Age	Gender	Occupation	Stay_In_Current_City_Years	Marital_Status	City_Category
0	1000001	0-17	F	10	2	0	A
4	1000002	55+	M	16	4+	0	C
5	1000003	26-35	M	15	3	0	A
6	1000004	46-50	M	7	2	1	B
9	1000005	26-35	M	20	1	1	A

Slika 4. Uzorak skupa jedinstvenih kupaca (autorski rad)

Koristeći *LabelEncoder* klasu iz scikit-learn Python modula, moguće je jednostavno pretvaranje kategoričkih i ostalih ne-numeričkih vrijednosti u numeričke vrijednosti (specifično u cjelobrojne vrijednosti počevši od nule). Na taj način se postiže ujednačavanje podataka, odnosno standardizacija.

LabelEncoder radi tako da pronalazi sve jedinstvene vrijednosti u stupcu te ih na jedinstven način supstituira cjelobrojnim vrijednostima počevši od nule (Pedregosa i ostali, 2011). Prema informacijama sa slike 1.(str. 10) moguće je saznati koji podaci trebaju biti konvertirani u numeričke vrijednosti. Od podataka koji su odabrani za model, četiri varijable su tipa podatka *object* a ostale dvije *int64*, prema tome, varijable tipa *object* potrebno je pretvoriti u numeričke.

Sljedeći programski isječak prikazuje primjer korištenja *LabelEncoder* klase.

```
labelEncoder = LabelEncoder()

skup_kupaca[stupac]= labelEncoder.fit_transform(skup_kupaca[stupac])
```

Na slici 5. prikazan je uzorak skupa podataka nakon obrade *LabelEncoder* klasom, može se primijetiti da su sve vrijednosti u skupu sada numeričke.

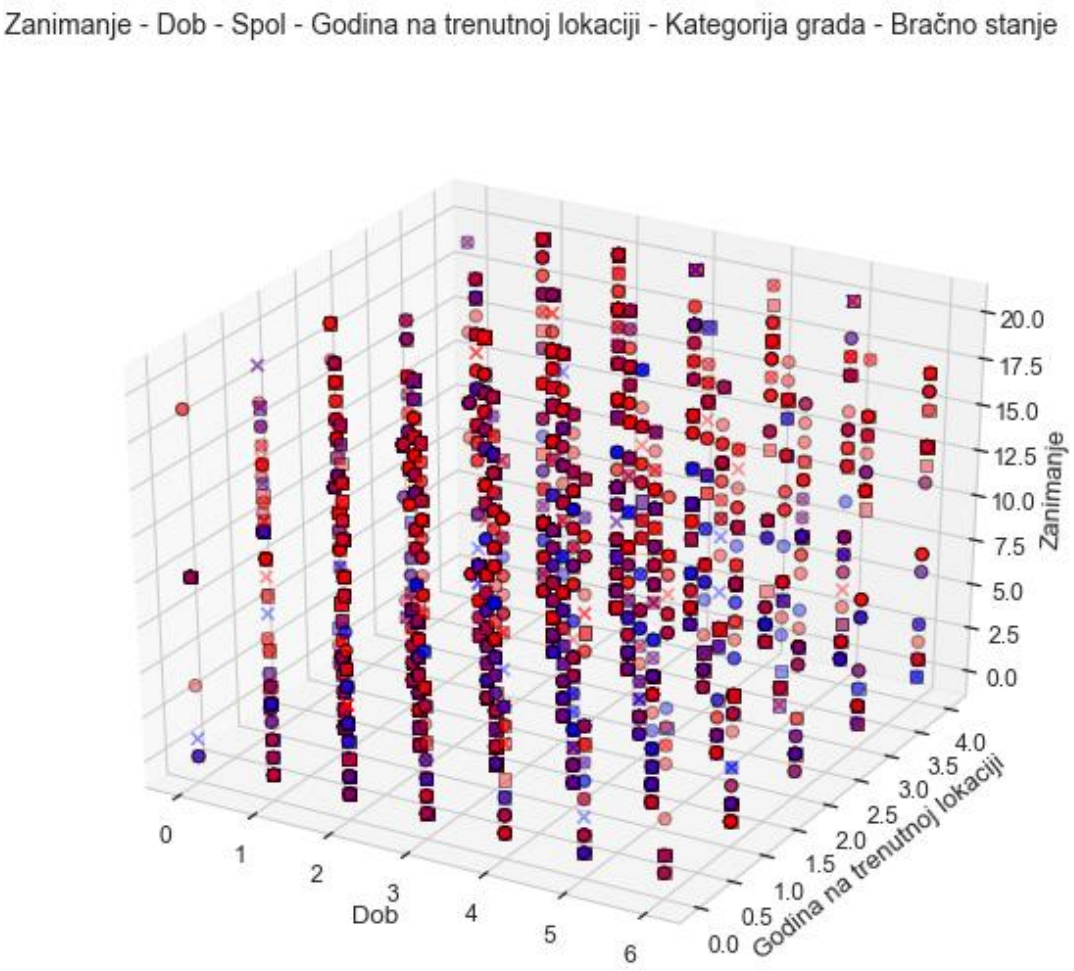
	Age	Gender	Occupation	Stay_In_Current_City_Years	Marital_Status	City_Category
0	0.0	0.0	10.0	2.0	0.0	0.0
4	6.0	1.0	16.0	4.0	0.0	2.0
5	2.0	1.0	15.0	3.0	0.0	0.0
6	4.0	1.0	7.0	2.0	1.0	1.0
9	2.0	1.0	20.0	1.0	1.0	0.0

Slika 5.Uzorak skupa nakon standardizacije vrijednosti (autorski rad)

Sad kada su odabrani atributi prema kojima će biti rađeno predviđanje i svi osim zadnjeg koraka pripreme podataka su izvršeni, podaci će još jednom biti vizualno prikazani. Podaci kakvi su prikazani na sljedećem grafikonu su onakvi kako će ih algoritam samo-organizirajuće mape analizirati, ali još ne-skalirani.

Za lakšu vizualizaciju, na sljedećem grafikonu nalaze se sve vrijednosti skupa (svih šest dimenzija), predstavljene na jednom grafikonu u tri dimenzije. Sa GRAFIKONA X moguće je primijetiti da je raspored podataka u prostoru prilično pravilno raspoređeni te da postoje jasne granice između grupa podataka. Takva svojstva su bila očekivana obzirom na to da su skoro svi atributi kupaca kategoričkog tipa. Iz tog razloga ranije spomenuti DBSCAN algoritam nije dobar odabir za kategorizaciju ovog skupa.

Na x, y i z osima grafikona nalaze se redom: dob kupca, koliko godina živi na trenutnoj lokaciji te zanimanje kojim se bavi. Crvene točke na grafu predstavljaju muškarce, plave točke žene. Točke koje su u obliku kvadrata predstavljaju kupce iz kategorije grada 'A', točke u obliku slova 'x' predstavljaju kupce iz 'B' kategorije grada te okrugle točke predstavljaju kupce iz 'C' kategorije grada. Za kraj, kupci koji nisu u braku prikazani su praznim/blijedim točkama dok su ostali kupci prikazani punim/obojenim točkama.



Graf 6. Prikaz skupa podataka u tri dimenzije (autorski rad)

4.3.3 Skaliranje vrijednosti

Posljednji korak pripreme podataka je skaliranje vrijednosti. Za to će biti korištena *StandardScaler* klasa iz *scikit-learn* Python modula. Skaliranje značajki koristi se kada se u skupu nalaze jako različiti podaci s vrijednostima koje se razlikuju po nekoliko redova veličine. Također i u slučaju da podaci nisu vrlo različitih vrijednosti skaliranje je obvezan korak pri radu s algoritmima koji mjere prostornu udaljenost podataka, kao što je SOM (Asaithambi, 2017).

StandardScaler računa standardnu vrijednost za svaku instancu podatka prema sljedećoj formuli:

$$Z = (x - u)/s$$

Pri čemu je Z standardna vrijednost instance, x je vrijednost instance, u predstavlja medijan skupa, a s je standardna devijacija (Pedregosa i ostali, 2011). Na slici 6. nalaze se prve dvije vrijednosti skupa nakon skaliranja.

```
array([[-1.74758177, -1.59248686,  0.29208147,  0.11035958, -0.85089772,
        -1.77387042],
       [ 2.25384501,  0.62794867,  1.24105762,  1.67129183, -0.85089772,
         0.84350482]])
```

Slika 6. Prikaz prva dva vektora kupca sa skaliranim vrijednostima

4.4 Primjena samo-organizirajuće mape na segmentaciju kupaca

U ovom poglavlju biti će pokazano modeliranje i treniranje Kohonenove mape. Za pisanje programskog koda korišten je Python modul *Somoclu* koji sadrži nekolicinu ugrađenih metoda za modeliranje i prikaz rezultata samo-organizirajuće mape.

Somoclu je razvijen u jeziku C++ s brzinom, efikasnošću i paralelnim izvođenjem na umu. Moguće je izvođenje na više jezgara procesora kao i na grafičkim karticama, realizirano pomoću OpenMP i CUDA programskih sučelja. *Somoclu* također nudi i vanjska programska sučelja za jezike Python, R i MATLAB (ovaj rad izrađen je korištenjem sučelja za Python). Zbog velike efikasnosti ovog programa moguća je i implementacija računalno zahtjevnih modela samo-organizirajućih mapa kao što je ESOM (Wittek, Gao, Lim, & Zhao, 2017).

Već je spomenuto da će se za kategorizaciju u ovom radu koristit specijalne vrsta samo-organizirajuće mape zvana ESOM (eng. *Emergent Self-Organizing Map*). Takve mape karakterizira velik broj neurona na izlaznom sloju mreže (tipično više od četiri tisuće) što dovodi do nastajanja struktura među podacima koje inače nisu primjetljive u običnim samo-organizirajućim mapama. Zbog navedenih svojstava, model se pokazao superiornijim pri rješavanju problema kategorizacije. Zbog svoje mogućnosti da drastično smanji dimenziju skupa podataka na dvodimenzionalnu mapu bez velikih gubitaka unutarnje strukture visoko - dimenzionalnih skupova, ESOM model tipično se koristi za vizualizaciju kompleksnih skupova podataka (Ultsch, 2005).

Za efikasnu implementaciju ESOM modela na problem kategorizacije, Vesanto & Alhoniemi (2000) predlažu korištenje postupka od dva koraka (eng. *two-level approach*). U

prvom koraku koristi se ESOM za kreiranje takozvanih prototipnih vektora (neuroni mreže) koji predstavljaju visoko-dimenzionalni prostor početnih podataka u dvodimenzionalnoj mapi. Zatim se, u drugom koraku, koristi neki od nenadziranih algoritama (hijerarhijski model ili k-sredina) za grupiranje prototipnih vektora, umjesto na početnom skupu podataka kao inače. Ovakav pristup u dva koraka čini cijeli proces računalno jeftinijim i bržim zato što je broj neurona mreže koje treba kategorizirat drastično manji nego broj instanci početnog skupa podataka.

Prvi korak kategorizacije, koji uključuje ESOM mrežu, služi kreiranju dodatnog sloja apstrakcije za daljnju analizu. Nastoji se postići vjerodostojna reprezentacija početnog skupa podataka, prikazana na dvodimenzionalnoj mapi koju čine izlazni neuroni ESOM mreže (Vesanto & Alhoniemi, 2000). Ovako prikazani podaci mogu se koristit samo za vizualnu analizu podataka temeljem u-matrica i ravnina komponenti (poglavlja 4.4.2 i 4.4.3), ali se mogu i grupirati primjenom nekog algoritma kategorizacije.

Primjenom algoritma kategorizacije nad neuronima ESOM mreže postiže se isti rezultat kao i kategoriziranje bilo kojeg drugog skupa podataka, a to su zasebne grupe podataka koje od drugih odjeljuju neka zajednička obilježja.

Kombiniranjem modela nadilaze se pojedine mane spomenutih algoritama čime se postižu bolji rezultati. U ove svrhe *Somoclu* omogućava podjelu neurona trenirane mreže s nekim od korisnički odabranih algoritama.

4.4.1 Odabir hiper-parametara

Prije treniranja SOM mape, potrebno je odrediti nekolicinu hiper-parametara (eng. *hyperparameters*) kao što su broj neurona mreže te broj epoha treniranja.

Broj izlaznih neurona mreže izrazito utječe na ishod učenja. SOM model s malim brojem izlaznih neurona, gdje je broj neurona jednak broju željenih kategorija, nije znatno bolji od računalno mnogo jeftinijih algoritama kao što su k-sredina. Za normalne SOM modele broj neurona procjenjuje se temeljem broja opservacija, sljedećom formulom (Tian, Azarian, & Pecht, 2014):

$$M \approx 5\sqrt{N}$$

Gdje je M aproksimirani broj neurona, a N broj opservacija. Broj opservacija dobije se množenjem broja redaka s množenjem broja stupaca skupa podataka. Da bi se odredio broj redaka i stupaca mape koju želimo dobiti korištenjem SOM modela, potrebno je korjenovati M vrijednost.

Rečeno je već da ESOM modeli sadrže velik broj neurona. Da bi se postigao efekt noviteta korištenjem ESOM modela, potrebno je imati nekoliko tisuća neurona, barem četiri tisuće prema (Ultsch & Morchen, 2005). U ovom radu koristit će se preporuka pisaca za iznad četiri tisuće neurona. Da se postigne isti broj neurona u redcima i stupcima uzima se $\sqrt{4000}$ za svaku od tih vrijednosti. Zaokruženo na veću cjelobrojnu vrijednosti broj redaka dakle iznosi šezdeset četiri te broj stupca isto toliko. S obzirom da je broj neurona iz navedenog članka samo smjernica za aproksimiranje veličine mape, daljnjim testiranjem autor je došao do zaključka da mapa sa šezdeset i šest neurona po retku i stupcu daje bolje rezultate.

Broj epoha treniranja je broj koliko je puta mreža obradila jedan cijeli skup podataka. Prilikom treniranja ESOM modela, broj epoha je nešto manji nego pri treniranju obične samo-organizirajuće mape. Prema Nocker, Morchen, & Ultsch (bez dat.) broj epoha za ESOM algoritam iznosi od deset do dvadeset pet. U ovom radu koristiti će se dvadeset epoha.

4.4.2 Treniranje samo-organizirajuće mape

Nakon što su odrađeni svi koraci pripreme podataka i biranja parametara za učenje, moguće je primijeniti ESOM model na skup podataka. U ranijem poglavlju opisana je teorijska pozadina samo-organizirajućih mapa, a u ovom poglavlju biti će pokazana implementacija.

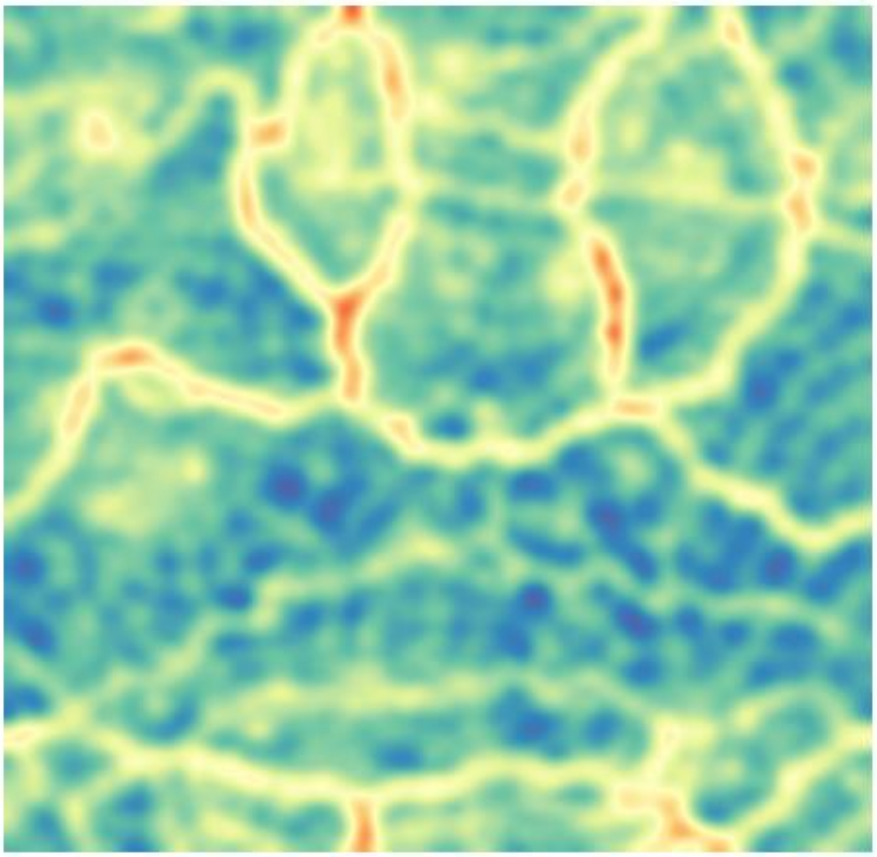
Sljedeći programski isječak prikazuje instanciranje `Somoclu` objekta, a zatim i treniranje novonastale mape.

```
broj_redaka = 66
broj_stupaca = 66
mapa = somoclu.Somoclu(broj_redaka, broj_stupaca, maptype = 'toroid')
mapa.train(data = kupci_vrijednosti, epochs=20)
```

U varijabli `mapa` pohranjen je objekt tipa `Somoclu`, koji kreira samo-organizirajuću mapu. Parametri koji se prosljeđuju objektu su: broj redaka, broj stupaca mape i tip mape. Kao tip mape uzet je toroidalan oblik, koji prema (Ultsch & Morchen, 2005) poboljšava rezultate treniranja, smanjenjem graničnih efekata (eng. *border effects*). Na kraju se u zadnjoj liniji nad objektom `mapa` poziva metoda `train()` koja započinje treniranje modela. Parametri koji se prosljeđuju metodi za treniranje su: skup podataka i broj epoha treniranja.

Po završetku treniranja napokon je moguće prikazati podatke organizirane na dvodimenzionalnoj mapi. Jedan primjer takve mape gradi se temeljem prosjeka udaljenosti vektora svakog neurona od svih susjednih neurona (tzv. 'U' vrijednost). Prikaz svih U-vrijednosti na istoj mapi zove se U-matrica, iz koje se može dobiti dojam gustoće podataka kao i naznake formacije kategorija (Ultsch, 2005).

Izvođenjem `metode view_umatix()` nad `Somoclu` objektom dobiva se prikaz U-matrice za trenirani model. Mjesta različitih gustoća, odnosno dobivene kategorije, odijeljene žutim/narančastim bojama. Na mapi se vide jasne granice među kategorijama, što je jasna naznaka uspješnog odijeljivanja podataka.



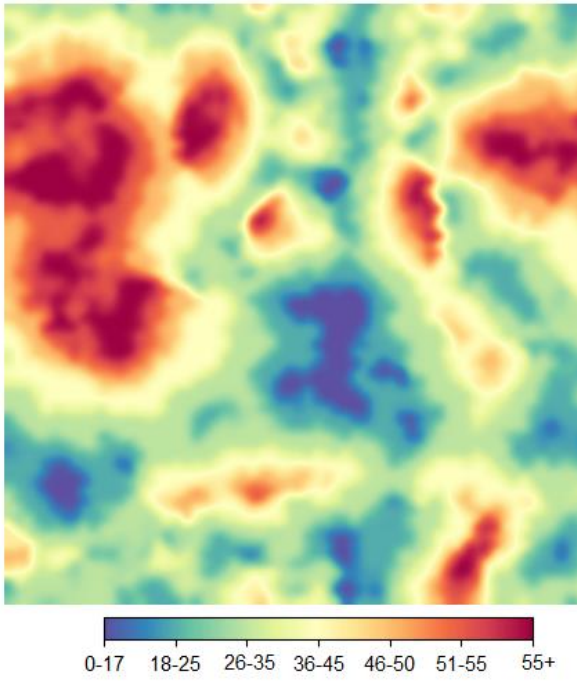
Slika 7. Prikaz U-matrice (autorski rad)

4.4.3 Analiza ravnina komponenti

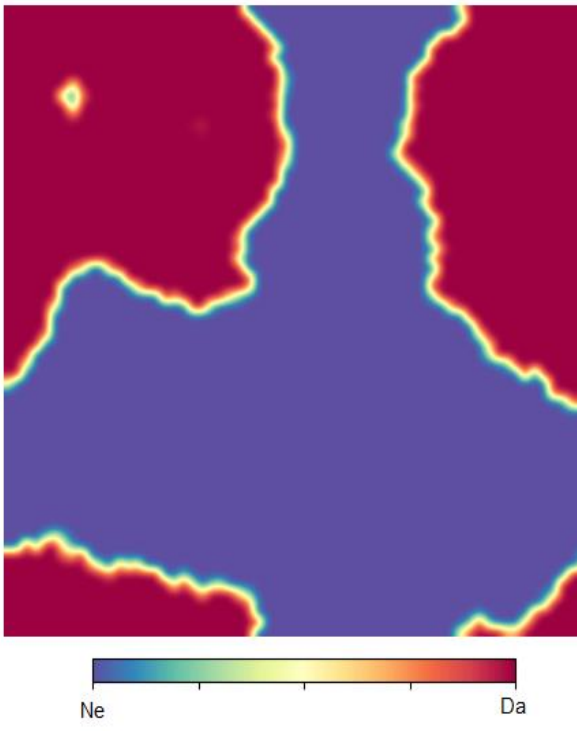
Ravnine komponenti (eng. *component planes*) su vizualne reprezentacije svake varijable, odnosno dimenzije skupa, na zasebnoj mapi. Promatranjem ravnina komponenti

mogu se uočiti obrasci u podacima te donijet zaključci o važnosti pojedine varijable na ishod kategorizacije. Naime, vrijedi da što su kategorije na mapi vizualno bolje odijeljene to su i podaci iz skupa bolje odijeljeni (Ghany & Solano, 2012).

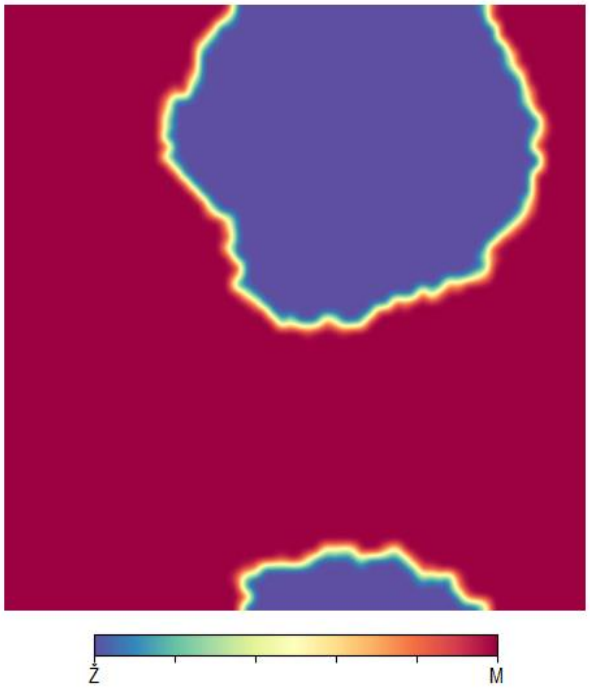
Na sljedećim slikama nalaze se ravnine komponenti za svaku varijablu koja je obrađena ESOM modelom. Iz slika se može vidjeti da su varijable spol (slika 10.), bračno stanje (slika 9.) i kategorija grada (slika 13.) vrlo dobro odijeljene, odnosno jasno se vide formirane kategorije.



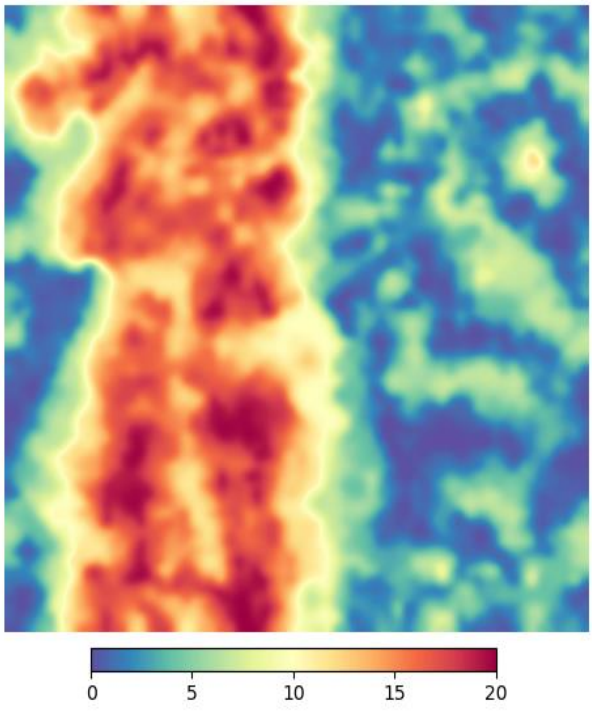
Slika 8. Prikaz ravnine komponente – Dob (autorski rad)



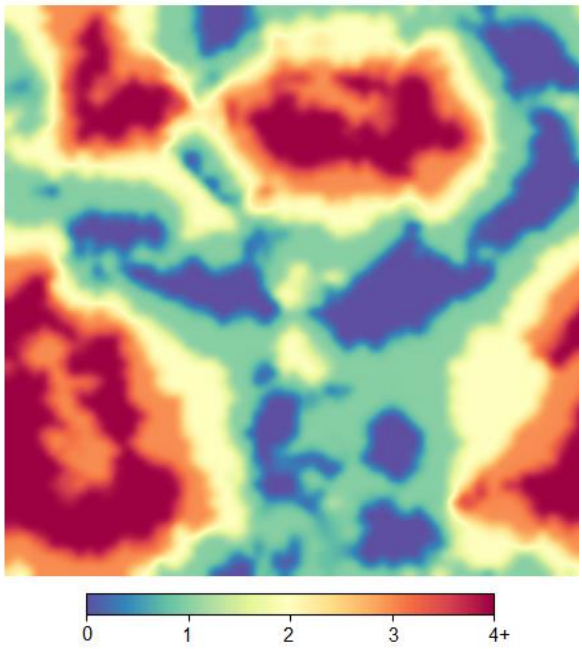
Slika 9. Prikaz ravnine komponente – Bračno stanje (autorski rad)



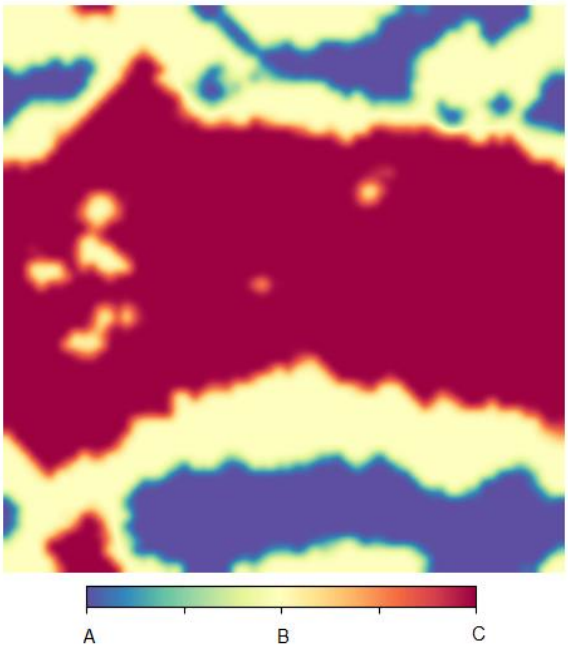
Slika 10. Prikaz ravnine komponente – Spol (autorski rad)



Slika 11.Prikaz ravnine komponente – Zanimanje (autorski rad)



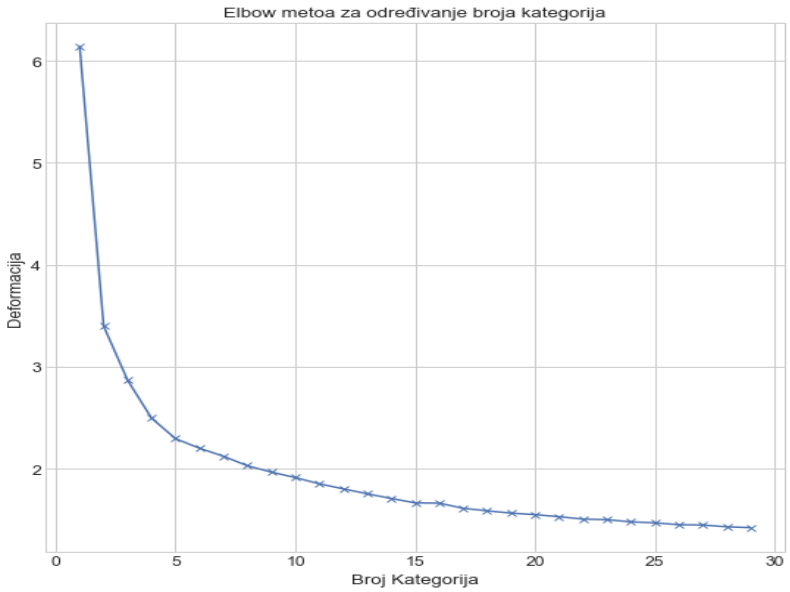
Slika 12. Prikaz ravnine komponente - Godina na trenutnoj lokaciji (autorski rad)



Slika 13.Prikaz ravnine komponente – Kategorija grada (autorski rad)

4.4.4 Određivanje broja kategorija

Prvi korak procesa je završen te je sada red da se trenirani neuroni podijele u konačan broj kategorija. Kako će se za podjelu neurona koristit K-sredina algoritam, potrebno je unaprijed odrediti broj kategorija. Za aproksimaciju broja kategorija korištena je *Elbow* metoda. Spomenuta metoda radi tako da se iterativno izvodi algoritam K-sredina na način da se u svakoj novoj iteraciji povećava broj kategorija s kojima algoritam radi. Cilj metode je doseći broj kategorije gdje dodavanje još jedne kategorije neće drastično smanjiti varijancu podataka (Bholowalia, 2014). Pravljenjem linijskog grafa gdje se na osi apscisa nalazi broj kategorija a na osi ordinata broj deformacije kategorija te traženjem točke gdje krivulja prestaje naglo padati, dobiva se aproksimiran broj kategorija. *Elbow* metoda nije egzaktna već samo aproksimira točan broj, kao rezultat se zato uzima neki broj kategorija u susjedstvu dobivenog rezultata. Na grafu 7. vidljivo je da krivulja prestaje naglo padati oko točke '5'. Testiranjem raznih vrijednosti zaključeno je da šest kategorija daje najbolje rezultate.

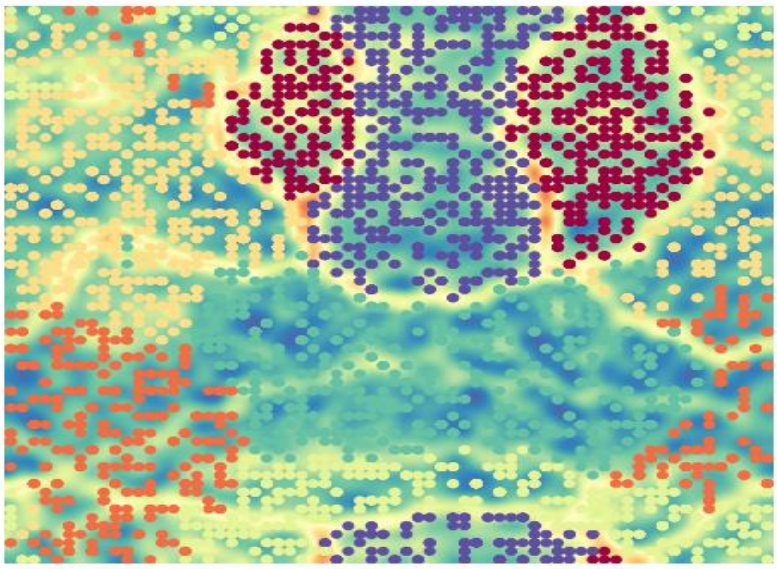


Graf 7. Prikaz linijskog grafa *Elbow* metode (autorski rad)

Sada kad je određen broj kategorija moguće je kategorizirati broj neurona ESOM modela. Sljedeći programski isječak pokazuje inicijalizaciju kategorija s k-sredina algoritmom postavljenim na šest kategorija.

```
mapa.cluster(algorithm=KMeans(n_clusters=6))
```

Time je završen proces segmentiranja kupaca. Pogledamo li sada U-matricu na slici 14., moguće je vidjeti sve kupce predstavljene točkama u zasebnim kategorijama. Svaka točka je jedan kupac, svaka kategorija opisana je jednom bojom.



Slika 14. Prikaz kupaca po kategorijama na U-matrici (autorski rad)

Zadnji korak u ovom potpoglavlju je dohvaćanje novonastale oznake svakog kupca i vraćanje podataka u ljudski čitljiv format. Nakon što su neuroni ESOM mape podijeljeni u optimalan broj kategorija, moguće je dobit oznaku kategorije za svakog kupca. Takav podatak uistinu je bitan zato što omogućava detaljnu analizu kupaca unutar kategorija što nadalje omogućava predviđanja temeljena dosadašnjim ponašanjem kupaca. Kategorija kupca dobiva se iteriranjem kroz svojstvo `cluster` objekta `mapa` na način da se za indekse koriste indeksi kupaca kako je i prikazano u sljedećem programskom isječku:

```
for kupac in range(kupci_vrijednosti.shape[0]):
      kategorije_lista.append(mapa.clusters[mapa.bmus[kupac,1],mapa.b
mus[kupac,0]])
```

Pretvaranjem podataka skupa natrag u ljudski čitljiv oblik dobiva se označeni skup podataka. Na slici 15. nalazi se uzorak skupa podataka koji sadrži novi stupac 'Kategorija_kupca'.

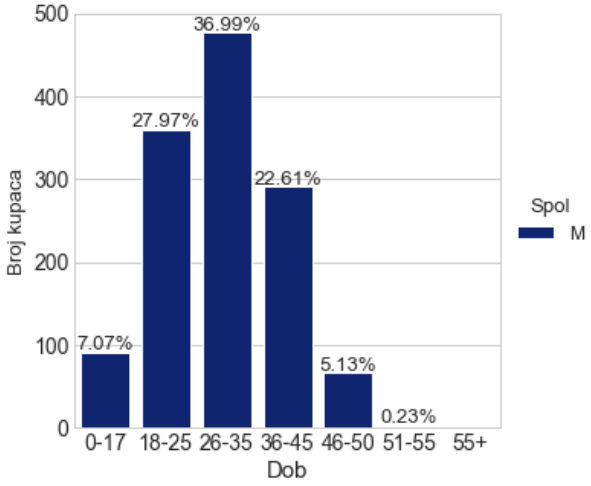
	Age	Gender	Occupation	Stay_In_Current_City_Years	Marital_Status	City_Category	Kategorija_kupca
0	0-17	F	10	2	0	A	0
1	55+	M	16	4+	0	C	3
2	26-35	M	15	3	0	A	2
3	46-50	M	7	2	1	B	3
4	26-35	M	20	1	1	A	2

Slika 15. Uzorak skupa sa dodanim stupcem 'Kategorija_kupaca' (autorski rad)

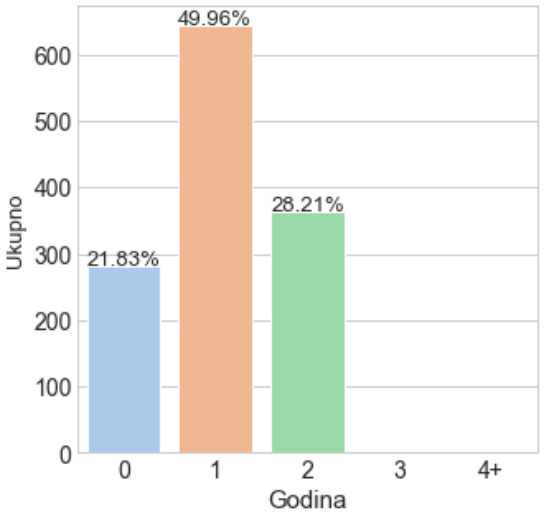
4.5 Analiza dobivenih rezultata

Jednom kad je svaki kupac dodijeljen u neku od kategorija moguće je raditi analize nad svakim pod-skupom te vidjeti po kojim je zajedničkim obilježjima model svrstao kupce u tu kategoriju. Svaka od šest kategorija sastoji se od jedinstvenih pod-skupova kupaca, u ovom poglavlju prikazan je pregled podataka unutar dvije zanimljivije od dobivenih kategorija.

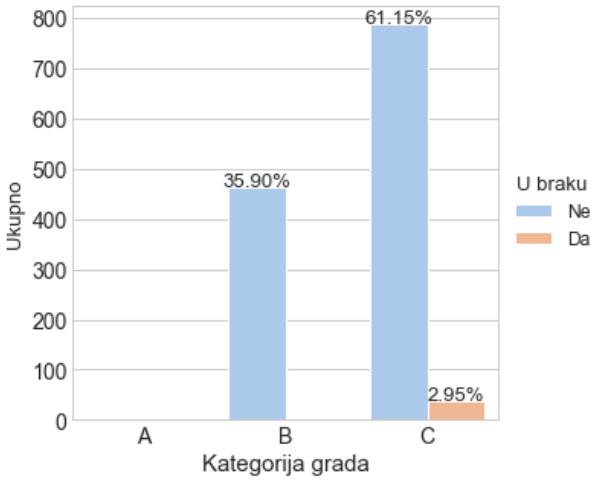
Na grafikonima 8., 9. I 10. mogu se vidjeti dijagrami koji opisuju demografsku strukturu Kategorije 2. Graf 8. prikazuje sadržaj pod-skupa prema dobi i spolu, može se primijetiti da se skup sastoji od isključivo muških osoba do 50 godina starosti. Zatim, graf 9., prikazuje koliko godina kupci žive na trenutnoj lokaciji, može se vidjeti da kupci u toj skupini žive do dvije godine na trenutnoj lokaciji. Na kraju, sa grafa 10. vidljivo je da svi kupci žive u B i C kategorijama grada te da 97% kupaca nije u braku.



Graf 8. Udio kupaca Kategorije 2 prema dobi i spolu (autorski rad)



Graf 9. Udio kupaca Kategorije 2 prema godinama življenja na trenutnoj lokaciji (autorski rad)

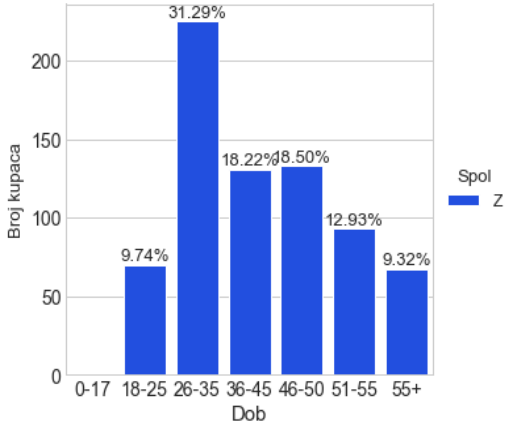


Graf 10. Udio kupaca Kategorije 2 prema kategoriji grada (autorski rad)

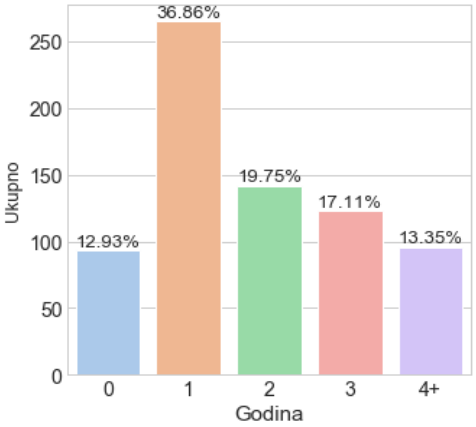
Na grafovima 11., 12. I 13. prikazani su podaci Kategorije 6. Sa grafa 11. može se vidjeti da se skup sastoji od isključivo punoljetnih ženskih osoba. Zatim na grafu 12. vidljivo je da su godine stanovanja na lokaciji distribuirane kao i na originalnom skupu (graf 1., str 11.). Odnosno, ova varijabla nije bila dovoljno utjecajna da bi temeljem nje model odijelio kupce. Na kraju, na grafu 13. vidi se da kupci iz Kategorije 6 žive u svim kategorijama gradova, većina u kategoriji C, te da su svi članovi skupine u braku.

Iako su ovdje prikazane samo dvije grupe, čak četiri od šest grupa kupaca sadrži isključivo muške osobe. Tome je razlog što skup podataka sadrži mnogo više primjera

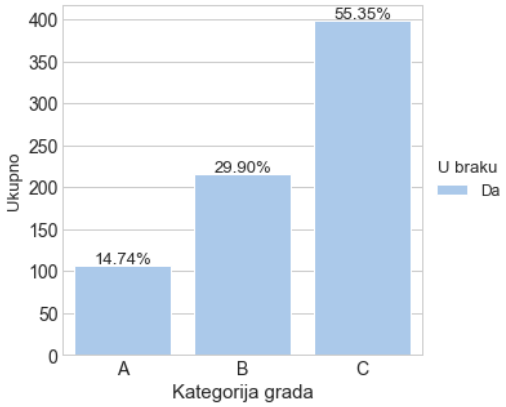
muških osoba (75% skupa čine muškarci) pa je algoritam imao više prilika za pronalaženje detaljnijih obrazaca na tom podskupu.



Graf 11. Udio kupaca Kategorije 6 prema dobi i spolu (autorski rad)



Graf 12. Udio kupaca Kategorije 6 prema godini življenja na trenutnoj lokaciji (autorski rad)



Graf 13.Udio kupaca Kategorije 6 prema kategoriji grada i bračnom stanju (autorski rad)

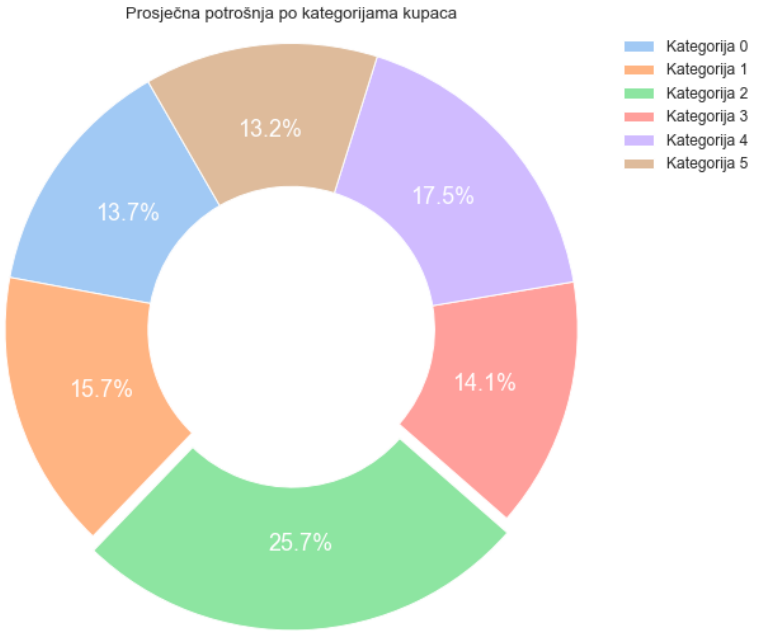
Jedan od pokazatelja vjerodostojnosti rezultata je činjenica da je model, sa stopostotnom točnošću, u zasebnu kategoriju (Kategorija 6) svrstao samo punoljetne ženske osobe koje su u braku. A sve maloljetne ženske osobe u drugu kategoriju (Kategorija 1), iako su te dvije kategorije po ostalim atributima vrlo slične. Prepoznavanje takvih obrazaca je upravo razlog korištenja ovakvih pristupa segmentaciji klijenata, iako je čovjeku takva podjela odmah uočljiva i razumna, valja imati na umu da algoritam inherentno ne poznaje takve pravilnosti (da su sve osobe u braku ujedno i punoljetne) te ih je pronašao isključivo

prepoznavanjem obrazaca u skupu podataka. Iako ovaj specifičan obrazac ne nosi veliku korist, uvelike pokazuje da algoritam pronalazi valjane uzorke među kupcima te da je moguće imati povjerenja da su i drugi obrasci valjani, a ne nasumično odabrani

4.6 Primjeri predviđanja za nove i postojeće kupce

Temeljem analiziranih podataka iz prošlog potpoglavlja moguće je donijeti razne zaključke o ponašanju sadašnjih kupaca te donijeti pretpostavke za nove kupce. Slijedi primjer preporuke proizvoda postojećem kupcu temeljem onoga što su kupili ostali kupci iz kupčeve kategorije.

Analizom pod-skupina kupaca moguće je saznati, na primjer, prosječnu potrošnju svake skupine. Ako postoji neka skupina kupaca koja u prosjeku više troši od ostalih skupina, te kupce se može smatrati jako vrijednima. Zatim, pravljenjem nekolicine upita nad originalnim skupom podataka moguće je pronaći koje su proizvode kupovali samo ti kupci. Temeljem tih podataka, kupcima iz skupine predlažu se proizvodi koje su kupovali ostali članovi iste skupine. Na dijagramu 14. prikazani su udjeli potrošnje svake skupine kupaca.



Graf 14. Udio prosječne potrošnje prema kategorijama kupaca (autorski rad)

Iz dijagrama se vidi da su u prosjeku najviše trošili kupci iz Kategorije 3 te da su zaslužni za čak 26% ukupne potrošnje. Temeljem te informacije, pretražuje se originalan skup podataka kako bi se pronašli proizvodi koje su kupovali samo kupci iz Kategorije 3. Sada se

neki od tih proizvoda mogu, u sklopu sustava za preporuku, preporučiti ostalim kupcima iz te skupine, odnosno onima koji još nisu kupili te proizvode. Na slici 16. nalazi se uzorak proizvoda koje su kupovali samo kupci Kategorije 3.

	Kategorija_kupca	Proizvod
0	2	P00054442
1	2	P00009642
2	2	P00161342
3	2	P00305442
4	2	P00107442

Slika 16. Uzorak jedinstvenih proizvoda u Kategoriji 3 (autorski rad)

Predviđanja za nove kupce odrađena su korištenjem jednostavnog klasifikacijskom algoritma - k-najbližih susjeda. Proces preporuke u ovom slučaju zamišljen je tako da se pri stvaranju novog računa, gdje je potrebno ostaviti podatke s kojima je model naučen, podaci novog kupca proslijede modelu k-najbližih susjeda koji je treniran na podacima iz dosadašnjeg skupa. Time se nastoji postići da će novi kupac biti smješten u skupinu gdje se nalaze njemu najsličniji kupci. Nakon što je novi kupac klasificiran, moguće je ponovno raditi iste upite kao što su ranije spomenuti da bi se dobili proizvodi iz kategorija tog kupca.

Model k-najbližih susjeda opisan je u ranijem poglavlju, a sada će biti prikazana njegova implementacija. Klasa za izgradnju modela sadržana je unutar *scikit-learn* Python modula (Pedregosa i sur, 2011.) te se implementira na sljedeći način:

```
knn_model = KNeighborsClassifier(n_neighbors=2)
knn_model.fit(znacajke,oznake)
```

Pri čemu je u varijablu `knn_model` pohranjen objekt modela te mu je proslijeđen broj 'susjeda' prema kojim će nova instanca biti klasificirana. Pozivanjem metode `fit()` nad objektom modela započinje treniranje, metodi se prosljeđuju parametri značajke i oznake. Značajke predstavljaju sve atribute o postojećim kupcima (dob, spol itd.) dok oznake predstavljaju broj kategorije kojoj pojedini kupac pripada, odnosno klasu kojoj pripada.

Nakon treniranja modela, pozivom metode `predict()` nad objektom modela te prosljeđivanjem atributa novog kupca u obliku niza, dobiva se predviđanje u koju klasu pripada novi kupac. Slijedi primjer novog kupca te predviđeni rezultat njegove kategorije. Novi kupci prosljeđuju se modelu u obliku niza na sljedeći način:

```
novi_kupci = np.asarray([['0-17','M','17','A','1','0']])
predvidanje = knn_model.predict(novi_kupci)
```

Varijabla `predvidanje` za danog kupca iznosi '2', odnosno, novi kupac klasificiran je u Kategoriju 3.

Ovime je pokazano da se kombinacijom nadziranih i nenadziranih modela strojnog učenja mogu dobiti jako vrijedne informacije za poslovanje. Kupci su uspješno segmentirani u šest zasebnih kategorija temeljem kojih se mogu raditi analize i praviti predviđanja za njihovo ponašanje.

Informacije dobivene ovakvom segmentacijom mogu se koristit prilikom sljedeće marketinške kampanje neke tvrtke čiji su kupci analizirani. Nadalje, dobivene informacije mogu poslužiti u direktnom marketingu ili se mogu povezati s CRM sustavom tvrtke kako bi se ostvarila još korisnija saznanja o poslovanju.

5. Zaključak

Predmet ovog rada bio je teorijski obraditi i prikazati primjenu modela strojnog učenja na predviđanje ponašanja potrošača. Taj cilj postignut je korištenjem tri različita modela. Model samo-organizirajuće mape korišten je otkrivanje unutarnje strukture među podacima te za prepoznavanje čovjeku teško uočljivih obrazaca u podacima. Zatim je modelom K-sredina obavljeno grupiranje neurona SOM mape u smislene kategorije iz kojih se mogu dobiti točne oznake grupe kojoj svaki kupac pripada. Temeljem tih informacija napravljena je analiza dvije od šest grupa kako bi se bolje upoznala struktura kategorija kupaca. Ovdje je pokazano da se u pod-skupinama nalaze kupci koji su prema nekim zajedničkim obilježjima svrstani u istu grupu. Na kraju su svi dobiveni podaci bili korišteni za donošenje predviđanja kupce. Za postojeće kupce pokazano je koje proizvode je moguće preporučiti unutar nekog sustava preporuka, temeljem prijašnjih ponašanja kupaca iz te kategorije. Predviđanje za nove kupce napravljeno je treniranjem modela k-najbližih susjeda nad podacima o postojećim kupcima. Takvim pristupom ostvareno je klasificiranje novog kupca u jednu od ranije dobivenih kategorija, čime je omogućeno daljnje predviđanje ponašanja kupca.

Programski kod pisan je u jeziku Python unutar Jupyter Notebook okružja. Za implementaciju modela korištene su vanjske biblioteke *Somoclu* i *Scikit-learn,* a za analizu podataka i pravljenje dijagrama Python biblioteke *Pandas*, *Matplotlib* i *Seaborn*.

U ovom radu pokazano je da se primjenom metoda strojnog učenja na segmentaciju potrošača mogu dobiti vrijedne informacije koje se dalje mogu koristit u poslovanju. Informacije dobivene ovim pristupom mogu se koristit u CRM aplikacijama ili vrijedno oruđe za analizu tržišta prije nove marketinške kampanje.

Popis literature

Asaithambi, S. (2017, prosinac). Why, How and When to Scale your Features. Preuzeto od https://medium.com/greyatom/why-how-and-when-to-scale-your-features-4b30ab09db5e

Asan, U., & Ercan, S. (2012). An Introduction to Self-Organizing Maps. U C. Kahraman (Ur.), *Computational Intelligence Systems in Industrial Engineering* (Sv. 6, str. 295–315). https://doi.org/10.2991/978-94-91216-77-0_14

Badea (Stroie), L. M. (2014). Predicting Consumer Behavior with Artificial Neural Networks. *Procedia Economics and Finance*, *15*, 238–246. https://doi.org/10.1016/S2212-5671(14)00492-4

Bean, R. (2018, rujan). The State of Machine Learning in Business Today. Preuzeto od https://www.forbes.com/sites/ciocentral/2018/09/17/the-state-of-machine-learning-in-business-today/#663c4b413b1d

Bhagat, A., Kshirsagar, N., Khodke, P., Dongre, K., & Ali, S. (2016). Penalty Parameter Selection for Hierarchical Data Stream Clustering. *Procedia Computer Science*, *79*, 24–31. https://doi.org/10.1016/j.procs.2016.03.005

Bholowalia, P. (2014). EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN. *International Journal of Computer Applications*, *105*(9), 8.

Bronsthein, A. (2017, travnja). A Quick Introduction to K-Nearest Neighbors Algorithm [Blog]. Preuzeto od https://blog.usejournal.com/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7

Dang, S. (2015). Performance Evaluation of Clustering Algorithm Using Different Datasets. *International Journal of Advance Research in Computer Science and Management Studies*, *3*(1), 167–173.

Darrin. (2016, kolovoz). Educational Research Techniques [Educational blog]. Preuzeto od K-Means Clustering website: https://educationalresearchtechniques.com/2016/08/05/k-means-clustering/

Euclidean Distance. (2005, rujan). *The Technical Whitepaper Series*, *6*, 26.

Galván, I. M., Valls, J. M., García, M., & Isasi, P. (2011). A lazy learning approach for building classification models. *International Journal of Intelligent Systems*, *26*(8), 773–786. https://doi.org/10.1002/int.20493

Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (1st izd.). O'Reilly Media, Inc. ©2017.

Ghany, M. L., & Solano, G. (2012). *Visualization of Multivariate Health Data Using Self-Organizing Maps*. 7.

Harrison, O. (2018, rujan). Machine Learning Basics with the K-Nearest Neighbors Algorithm. Preuzeto od https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

Kilitcioglu, D. (2018, listopad). Hierarchical Clustering and its Applications [Blog post]. Preuzeto od https://towardsdatascience.com/hierarchical-clustering-and-its-applications-41c1ad4441a6

Le, J. (2019, ožujak). An Introduction to Big Data: Data Cleaning. Preuzeto od https://medium.com/cracking-the-data-science-interview/an-introduction-to-big-data-data-cleaning-a238725a9b2d

Mirošević, I. (2016). *Algoritam k-sredina*. 8.

Nocker, M., Morchen, F., & Ultsch, A. (bez dat.). *An algorithm for fast and reliable ESOM learning*. 6.

Pazzani, M. J., & Billsus, D. (2007). Content-Based Recommendation Systems. U P. Brusilovsky, A. Kobsa, & W. Nejdl (Ur.), *The Adaptive Web* (Sv. 4321, str. 325–341). https://doi.org/10.1007/978-3-540-72079-9_10

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Cournapeau, D. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 6.

Raghupathi, K. (2018, ožujak). 10 Interesting Use Cases for the K-Means Algorithm. Preuzeto od https://dzone.com/articles/10-interesting-use-cases-for-the-k-means-algorithm

Sari, J. N., Nugroho, L. E., Ferdiana, R., & Santosa, P. I. (2016). Review on Customer Segmentation Technique on Ecommerce. *Advanced Science Letters*, *22*(10), 3018–3022. https://doi.org/10.1166/asl.2016.7985

Shaikh, R. (2018, listopad). Feature Selection Techniques in Machine Learning with Python. Preuzeto od https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e

Sublett, D. (2018, srpanj). Beginner's Recommendation Systems with Python. Preuzeto od https://towardsdatascience.com/beginners-recommendation-systems-with-python-ee1b08d2efb6

Tian, J., Azarian, M. H., & Pecht, M. (2014). Anomaly Detection Using Self-Organizing Maps-Based K-Nearest Neighbor Algorithm. *Scientia Iranica*, *18.*, 9. https://doi.org/10.1016/j.scient.2011.08.025

Top 8 Business Benefits of Machine Learning. (2018, kolovoz). Preuzeto od https://sailotech.com/blog/top-8-business-benefits-of-machine-learning/

Ultsch, A. (2005). *CLUSTERING WIH SOM: U*C*. 9.

Ultsch, A., & Morchen, F. (2005). *ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM*. 7.

Vesanto, J., & Alhoniemi, E. (2000). Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, *11*(3), 586–600. https://doi.org/10.1109/72.846731

Wagstaff, K., Cardie, C., Rogers, S., & Schroedl, S. (2001). Constrained K-means Clustering with Background Knowledge. *Proceedings of the Eighteenth International Conference on Machine Learning*, 8.

Wittek, P., Gao, S. C., Lim, I. S., & Zhao, L. (2017). An Efficient Parallel Library for Self-Organizing Maps. *Journal of Statistical Software*, *78*(9). https://doi.org/10.18637/jss.v078.i09

Yang, X.-S., Deb, S., & Fong, S. (2011). Accelerated Particle Swarm Optimization and Support Vector Machine for Business Optimization and Applications. U S. Fong (Ur.), *Networked Digital Technologies* (Sv. 136, str. 53–66). https://doi.org/10.1007/978-3-642-22185-9_6

Popis slika

Popis grafikona